Problem Set #1

✓ **5/5** 得分（100%）

✓ 1 / 1 分

1.

We are given as input a set of n requests (e.g., for the use of an auditorium), with a known start time s_i and finish time t_i for each request i. Assume that all start and finish times are distinct. Two requests *conflict* if they overlap in time --- if one of them starts between the start and finish times of the other. Our goal is to select a maximum-cardinality subset of the given requests that contains no conflicts. (For example, given three requests consuming the intervals $[0, 3]$, $[2, 5]$, and $[4, 7]$, we want to return the first and third requests.) We aim to design a greedy algorithm for this problem with the following form: At each iteration we select a new request i, including it in the solution-so-far and deleting from future consideration all requests that conflict with i.

Which of the following greedy rules is guaranteed to always compute an optimal solution?

○ At each iteration, pick the remaining request with the fewest number of conflicts with other remaining requests (breaking ties arbitrarily).

○ At each iteration, pick the remaining request with the earliest finish time.

▲

Let R_j denote the requests with the j earliest finish times. Prove by induction on j that this greedy algorithm selects the maximum-number of non-conflicting requests from S_j.

○ At each iteration, pick the remaining request which requires the least time (i.e., has the smallest value of $t_i - s_i$) (breaking ties arbitrarily).

○ At each iteration, pick the remaining request with the earliest start time.

✔ 1 / 1 分

2.

We are given as input a set of n jobs, where job j has a processing time p_j and a deadline d_j. Recall the definition of *completion times* C_j from the video lectures. Given a schedule (i.e., an ordering of the jobs), we define the *lateness* l_j of job j as the amount of time $C_j - d_j$ after its deadline that the job completes, or as 0 if $C_j \le d_j$. Our goal is to minimize the maximum lateness, $\max_j l_j$.

Which of the following greedy rules produces an ordering that minimizes the maximum lateness? You can assume that all processing times and deadlines are distinct.

○ Schedule the requests in increasing order of processing time p_j

○ Schedule the requests in increasing order of the product $d_j \cdot p_j$

○ None of the other answers are correct.

○ Schedule the requests in increasing order of deadline d_j

▲

Proof by an exchange argument, analogous to minimizing the weighted sum of completion times.

3.

Consider an undirected graph $G = (V, E)$ where every edge $e \in E$ has a given cost c_e. Assume that all edge costs are positive and distinct. Let T be a minimum spanning tree of G and P a shortest path from the vertex s to the vertex t. Now suppose that the cost of every edge e of G is increased by 1 and becomes $c_e + 1$. Call this new graph G'. Which of the following is true about G'?

○ T must be a minimum spanning tree but P may not be a shortest s-t path.

正确回答

The positive statement has many proofs (e.g., via the Cut Property). For the negative statement,

think about two different paths from s to t that contain a different number of edges.

○ T is always a minimum spanning tree and P is always a shortest s-t path.

○ T may not be a minimum spanning tree and P may not be a shortest s-t path.

○ T may not be a minimum spanning tree but P is always a shortest s-t path.

4.

Suppose T is a minimum spanning tree of the connected graph G. Let H be a connected induced subgraph of G. (I.e., H is obtained from G by taking some subset $S \subseteq V$ of vertices, and taking all edges of E that have both endpoints in S. Also, assume H is connected.) Which of the following is true about the edges of T that lie in H? You can assume that edge costs are distinct, if you wish. [Choose the strongest true statement.]

○ For every G and H, these edges form a spanning tree (but not necessary minimum-cost) of H

○ For every G and H and spanning tree T_H of H, at least one of these edges is missing from T_H

⊙ For every G and H, these edges are contained in some minimum spanning tree of H

▲

Proof via the Cut Property (cuts in G correspond to cuts in H with only fewer crossing edges).

○ For every G and H, these edges form a minimum spanning tree of H

✔ 1 / 1 分

5.
Consider an undirected graph $G = (V, E)$ where edge $e \in E$ has cost c_e. A *minimum bottleneck spanning tree* T is a spanning tree that minimizes the maximum edge cost $\max_{e \in T} c_e$. Which of the following statements is true? Assume that the edge costs are distinct.

○ A minimum bottleneck spanning tree is always a minimum spanning tree but a minimum spanning tree is not always a minimum bottleneck spanning tree.

⊙ A minimum bottleneck spanning tree is not always a minimum spanning tree, but a minimum spanning tree is always a minimum bottleneck spanning tree.

▲

For the positive statement, recall the following (from correctness of Prim's algorithm): for every edge e of the MST, there is a cut (A, B) for which e is the cheapest one crossing it. This implies that every other spanning tree has maximum edge cost at least as large. For the negative statement, use a triangle with one extra high-cost edge attached.

○ A minimum bottleneck spanning tree is not always a minimum spanning tree and a minimum spanning tree is not always a minimum bottleneck spanning tree.

○ A minimum bottleneck spanning tree is always a minimum spanning tree and a minimum spanning tree is always a minimum bottleneck spanning tree.